Reinforcement Learning: An Introduction

Solutions: Chapter 3

Mrinank Sharma

March 28, 2020

Exercise 3.1 MDP Examples

1. **Uber Example**. Uber needs to choose which cars to allocated to which people requesting a ride. State is information about the passenger, other relevant information (e.g., time of day, previous knowledge about the area), information about the cars which are available. Action would be which car to allocate to the passenger. Reward could be a number of different things, including: a waiting time penalty, customer satisfaction reward, maybe some other things to.
2. **Youtube Example**. Youtube autoplay needs to choose the next video for you to watch. The state is it's knowledge about the viewer as well as other relevant things e.g., perhaps time of day. Reward will the be total time that the viewer spends on youtube (it could be given incrementally though).
3. **Me choosing a book**. I need to choose the next book that I want to read. The state is my knowledge about myself e.g., my budget, my time budget. The action space is the available books that I could purchase. The reward is my satisfaction from reading the book.

Not entirely sure that these are that different from each other, or that stretched. We could formulate supervised learning as an RL problem. The action is a prediction to be made for a person, the state is information about that perform and the reward is based on some measure of how close the classification (or regression) is to the true value. I guess this would work.

Exercise 3.2: MDP Insufficiency

Some issues with MDPs could be that:

- **Partial Observability**. We might know what the true Markov state is but simply be unable to measure it.
- **Multiple rewards**. We might want to maximise multiple things, likely involving a tradeoff. I suppose we can define some new arbitary reward function, but is this exactly what we want?
- **Non-Stationary Dynamics**. It's assumed that the p function remains the same (as far as I can tell). However, it could change over time, for example, humans reward functions change over a course of a lifetime.
- **Can we formulate all behaviour we want as reward maximisation?** Not sure I agree with this point (VNM Rationality). Some people satisfice, but I suppose that means the reward function we have down is wrong. If they are rational, they should be maximising something.
- **Self-Direction**. In real life, people have internal rewards i.e., they choose what to pursure. Indeed, life is not entirely directed as people have a lot of freedom. In the MDP, we've assumed that this reward function is entirely external.
- **Multiple Agents**. This is linked to non-stationarity, but many situations have multiple agents. We need game theoretical tools here most likely. Probably can be modelled as a changing environment.

Exercise 3.3: Drawing the Line

I think the choice of action depends on precisely the problem we are trying to solve with RL. For example, if we want to solve where to go to have a good weekend, pick the high level actions. If we take where we are trying as fixed, the actions are then moreso how to drive. This gives the sense that the problems that we want to solve are hierarchical - we want to be able to plan high level behaviour as well as learn the lower level skills enabled to execute them.

We might prefer one location over another because it makes the RL problem easier. For example, we could draw the agent box very small and make the actions the electrical signals that are being sent out. This seems like it could be a difficult problem to solve. I'm not sure how compelling this argument is though.

Actions should always be things that can be controlled by the agent.

Exercise 3.4: p functions

Please see Table 1 for the dynamics table.

Table 1: Dynamics Table

s	a	s'	r	$p(s', r\|s, a)$
high	search	high	$r_{\texttt{search}}$	α
high	search	low	$r_{\texttt{search}}$	$1-\alpha$
high	wait	high	$r_{\texttt{wait}}$	1
low	wait	low	$r_{\texttt{wait}}$	1
low	recharge	high	0	1
low	search	low	$r_{\texttt{search}}$	β
low	search	high	-3	$1-\beta$

Exercise 3.5: Episodic Alteration

Using the notation of $\mathcal{S}$ to include all states and $\mathcal{S}^+$ to include only non-terminal states, we can write:

$$\sum_{s' \in S} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \text{for all } s \in \mathcal{S}^+, a \in \mathcal{A}(s). \tag{1}$$

However, for the terminal states:

$$p(s' = s, r = 0|s \in \mathcal{S}, s \notin \mathcal{S}^+) = 1, \tag{2}$$

i.e., if we are in a terminal state, the next state is the same and there is no reward.

Not sure if the above equation is strictly necessary to be honest!

Exercise 3.6: Continuing and Episodic Tasks

In the episodic case, the return is $G_t = -\gamma^k$ where k is the number of timesteps til the cart falls.

In the continuing case, the cart will fall an infinite number of times, not just once! Hence $G_t = -\gamma^{k_1} - \gamma^{k_2} \dots$ where k_1 is time to the next failure, k_2 is time to the failure after that, e.t.c.,

Exercise 3.7: Faulty Reward Signal

We haven't communicated that we want the robot to escape the maze *as quickly as possible*. Without discounting, the reward is the same for every episode which eventually finds the maze exit. We need some discounting i.e, the reward needs to get smaller if the robot took more timesteps.

Note that if we assume that every episode goes on indefinitely, the only way termination happens is if the robot reachs the end. Thus, every single completed episode has the exact same reward. There is no way for a learner to discovered what the desired task actually is.

Exercise 3.8: Episodic Return

Use the recursive formula. $G_5 = 0, G_4 = 2, G_3 = 3 + 0.5 \cdot 2 = 4, G_2 = 6 + 0.5 \cdot 4 = 8, G_1 = 2 + 0.5 \cdot 8 = 6, G_0 = -1 + 0.5 \cdot 6 = 2.$

Exercise 3.9: Continuing Return

$G_1 = 7 + 0.9 \cdot 7 + 0.9^2 \cdot 7 + \dots = \frac{7}{1-0.9} = 70.$ $G_0 = 2 + 0.9 \cdot 70 = 65$

Exercise 3.10: Infinite Series Proof

Note: in my version of the book, this exercise didn't make sense - it's not clear what inequalty it's refering to. Instead, I proved (3.10).

$$\begin{aligned} G_t &= 1 + \gamma + \gamma^2 + \dots. \\ &= 1 + \gamma G_t \\ &= \frac{1}{1-\gamma} \end{aligned} \tag{3}$$

Exercise 3.11

$$\mathbb{E}[R_{t+1}] = \sum_{r\in\mathcal{R}} \sum_{a\in\mathcal{A}(s)} \sum_{s_{t+1}\in\mathcal{S}} p(s', r|s, a)\pi(a|s) \tag{4}$$

Exercise 3.12

$$v_\pi(s) = \sum_{a\in\mathcal{A}(s)} \pi(a|s)\, q_\pi(s, a) \tag{5}$$

Exercise 3.13

$$q_\pi(s, a) = \sum_{s'\in\mathcal{S}, r\in\mathcal{R}} p(s', r|s, a)\big[r + \gamma v_\pi(s')\big] \tag{6}$$

Exercise 3.14

The value of the centre, by the recursive relationship ought to be:

$$\frac{0.9}{4}[2.3 + 0.7 - 0.4 + 0.4] = 0.675 \simeq 0.7(1 \text{ d.p.}) \tag{7}$$

Exercise 3.15: Shifting Rewards

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{8}$$

Consider the shifted return.

$$\begin{aligned} \hat{G}_t &= \sum_{k=0}^{\infty} \gamma^k \hat{R}_{t+k+1} \\ &= \sum_{k=0}^{\infty} \gamma^k [R_{t+k+1} + c] \\ &= G_t + \frac{c}{1-\gamma} \end{aligned} \tag{9}$$

The value function is the expected return given an initial state, and thus this function is also shifted. It is only the difference between rewards that makes a difference.

Exercise 3.16: Episodic Shifting Rewards

In this case, it will make a difference. Consider an episodic task with a reward of -1 on each time step and a reward of $+5$ for finishing. This creates the behaviour of trying to minimise the time taken before finishing. Then, instead consider a reward of $+1$ on each timestep and a reward of $+7$ for finishing. Maximising this reward entails *taking as long as possible to complete the task - certainty* ***not*** *what we wanted*.

The different in episodic tasks is that we don't know how long the episode was going to last. If the episode was a fixed length, this problem would go away and it wouldn't make a difference (we can see this by imagining the above sum - we would add constant to each of the value functions.)

Exercise 3.17: Bellman for Action-Value Functions

$$\begin{aligned} q\pi(s, a) &\triangleq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a] \\ &= \sum_{r\mathcal{R}} p(s', r|s, a)[r + \gamma G_{t+1}] \\ &= \sum_{r\in\mathcal{R}} p(s', r|s, a)[r + \gamma \sum_{a'\in\mathcal{A}(s')} q_\pi(s', a')], \end{aligned} \tag{10}$$

which matches the backup diagram provided in the book.

Exercise 3.18: Value Functions and Action-Value Functions

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi(q_\pi(s,a)) \\ &= \sum_{a\in\mathcal{A}(s)} \pi(a|s) q_\pi(s,a) \end{aligned} \tag{11}$$

Exercise 3.19: Value Functions and Action-Value Functions

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi(G_t|S_t=a, A_t=a) \\ &= \mathbb{E}_\pi(R_{t+1}+\gamma G_{t+1}|S_t=a, A_t=a) \\ &= \sum_{s'\in\mathcal{S}, r\in\mathcal{R}} p(s',r|s,a)[r+\gamma v_\pi(s')] \\ &= \mathbb{E}_p[R_{t+1}+\gamma v_\pi(S_{t+1})|S_t=a, A_t=a] \end{aligned} \tag{12}$$

Exercise 3.20: Golf-Example

The optimal value function is a merge between the function for putting and the function using the driver. It has the same contours as the driver outside the green. On the green, it has the contours from the putter (or the entire green has value -1).

Exercise 3.21: Golf Part 2

This looks the exact same as the value function for putting.

Exercise 3.22: Discounting

With $\gamma = 0$, we only care about the reward one step ahead, so always so left.

With $\gamma = 0.5$, the value for going left is 1 and the value for going right is $2 \cdot 0.5 = 1$ so choose artibarily between the two.

With $\gamma = 0.9$, the value for going right is $2 * 0.9 > 1$, so always go right.

Exercise 3.23: Robot Returns

$$q_*(h,w) = r_{\texttt{wait}} + \gamma \max_a q_*(h,a) \tag{13}$$

$$q_*(h,s) = \alpha[r_{\texttt{search}} + \gamma \max_a q_*(h,a)] + (1-\alpha)[r_{\texttt{search}} + \gamma \max_a q_*(l,a)] \tag{14}$$

$$q_*(l,s) = \beta[r_{\texttt{search}} + \gamma \max_a q_*(l,a)] + (1-\beta)[-3 + \gamma \max_a q_*(h,a)] \tag{15}$$

$$q_*(l,r) = \gamma \max_a q_*(h,a) \tag{16}$$

$$q_*(l,w) = r_{\texttt{wait}} + \gamma \max_a q_*(l,a) \tag{17}$$

these can be solved in principle, but it isn't that nice with those maxima.

Exercise 3.24: Gridworld

We first find γ.

$$v_*(A') = 0 + \gamma v_*(\text{box above A}) \Rightarrow \gamma = 0.9 \tag{18}$$

Now,

$$v_*(A) = R_{AA'} + \gamma v_*(A') = 24.4 \tag{19}$$

Exercise 3.25 - 3.28: Bellman Optimality

$$v_*(s) = \max_{a\in\mathcal{A}(s)} q_*(s,a). \tag{20}$$

$$q_*(s,a) = \sum_{s',r'} p(s',r|s,a)[r + \gamma v_*(s')]. \tag{21}$$

$$\pi_*(a|s) = \arg\max_a q_*(s,a), \tag{22}$$

where ties are broken arbitarily.

$$\pi_*(a|s) = \arg\max_a \sum_{s',r'} p(s',r|s,a)[r + \gamma v_*(s')], \tag{23}$$

directly lifted from a previous solution.

Exercise 3.29: Rewriting Recursions

$$v_\pi(s) = \sum_{a\in\mathcal{A}(s)} \pi(a|s)[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|a,s)v_\pi(s')] \tag{24}$$

$$v_*(s) = \max_{a\in\mathcal{A}(s)} r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|a,s)v_*(s') \tag{25}$$

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|a,s) \sum_{a'\in\mathcal{A}(s')} \pi(a|s)q_\pi(s',a') \tag{26}$$

$$q_*(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) \max_{a'\in\mathcal{A}(s')} q_*(s',a') \tag{27}$$

A slightly different formulation, though I think I prefer the versions with full state dynamics.